

- I
- a) F f) F, permite um ordenamento que é tão bom como o binário, mas com uma complexidade específica
 - b) F g) F
 - c) V h) F, se não há um dígito em comum, então não há uma comparação entre os dígitos
 - d) F i) F, é ao contrário
 - e) V j) F, este símbolo não ordena e ordena por estes símbolos e intervalos

II

Uma vez que cada nó contém os nomes ordenados alfabeticamente, podemos utilizar o algoritmo binary search em cada nó para encontrar o próximo nó ou localizações do nome 'Katz'.

Comçando no nó da raiz, como alfabeticamente 'Katz' < 'Mozart', procuramos os nós à esquerda de 'Mozart'.

Nesse nó, como alfabeticamente 'Katz' > 'Gold', vamos para o nó à direita de 'Gold'.

Procuramos nesse nó, finalmente encontramos o nome Katz.

A complexidade assintótica de pesquisa é $O(\log n)$. A árvore é balanceada, o que contribui para a performance da árvore.

Em cada nó da árvore podemos aplicar um binary search de nós a tornar o processo mais eficiente, desde que, em cada nó, a lista está ordenada.

III

A propriedade $A(i)$ não atômica, consistência, isolamento e durabilidade.

A atômica refere-se ao facto de as transações serem completamente efetivas ou não serem efetivas de todo.

A consistência refere-se ao facto de as transações "bons" a base de dados de um estado consistente para outro estado consistente.

O isolamento refere-se ao facto de as transações concorrentes não terem acesso a dados inconsistentes na base de dados. Isto é, cada transação é executada independentemente das outras.

Durabilidade refere-se ao facto de após uma transação ter sido commitada, as alterações na base de dados não forem perdidas, mesmo que haja alguma falha de software ou hardware.

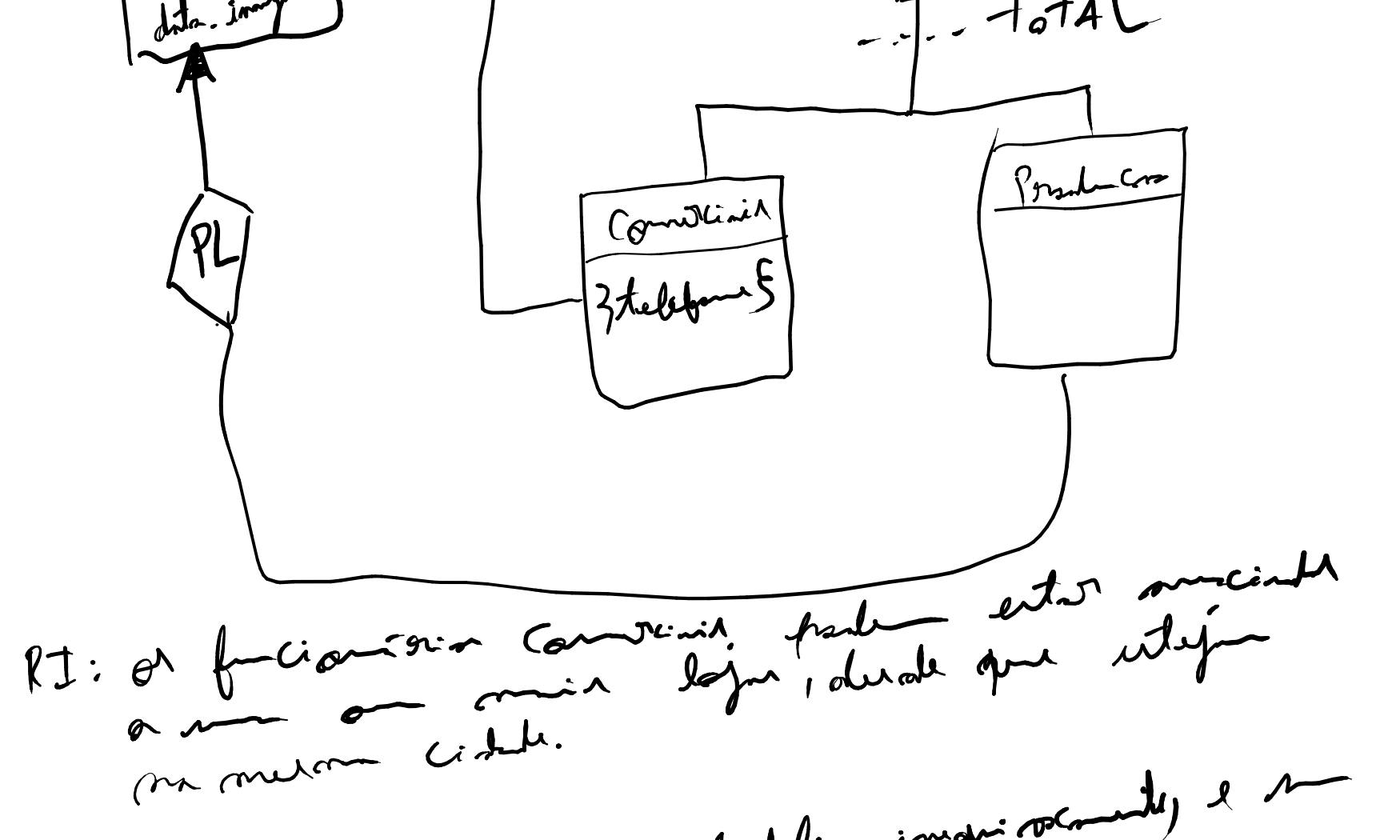
Estas propriedades são muito importantes em uma base de dados relacional, pois garantem a consistência dos dados em caso de falha de software ou hardware, falta de energia, transações concorrentes a tentarem alterar os mesmos dados não perdendo, mesmo que haja alguma falha de software ou hardware.

IV

Superchave é um conjunto de atributos de uma relação que identifica inequivocamente qualquer instância. Pode haver várias superchaves em uma relação.

Chave candidata é um superchave mínima, isto é, com o menor número de atributos. Pode haver várias chaves candidatas em uma relação.

(Chave primária corresponde a chave candidata escolhida para ser utilizada como chave primária. Em cada relação só pode existir uma chave primária, diferentemente das candidatas).



RI: as funções devem poder estar associadas a um ou mais locais, desde que estejam no mesmo cidade.

VI

E1 tem um a, que o identifica inequivocamente, e um b.

E1 pode ter um E2 ou E3 (disjunção).

Por E3 deve ser omitido um link de C.

E5 tem um a, que o identifica inequivocamente, e um b.

E4 tem um a e um b que juntos o identificam inequivocamente.

A cada E3 está associado um E4.

A cada E2 pode estar associado um ou mais E5.

Na associação de E2 com E5 há um atributo C.

$E1(a, b) \rightarrow E3C(a, b)$
 $E2(a) \rightarrow E4(a, b)$
 $FK(a) \rightarrow E1$
 $E3(a, w) \rightarrow E4(a, w, C)$
 $FK(a) \rightarrow E1$
 $FK(w) \rightarrow E4$
 $FK(a) \rightarrow E2$
 $FK(w) \rightarrow E5$
 $RI: E1 \text{ e } E2 \text{ ou } E3 \text{ ou } E4 \text{ ou } E5$
 (disjunção no total)

VII

a) $\Pi_{nome} (\sigma_{salario > 2000} (professor))$

b) $\Pi_{nome, nome_dept} (\sigma_{(professor \times departamento) \wedge nome_dept = nome_departamento} (professor \times departamento))$

c) $\Pi_{nome, salario} (\sigma_{(professor \times departamento) \wedge nome_dept = nome_departamento} (professor \times departamento))$

d) $\Pi_{nome_dept} (\sigma_{(count(nome_func) \neq count(nome_func), max(salario) \neq max(salario))} (professor))$

VIII

a) select nome from professor where salario > 2000;

b) select nome, nome_dept from professor, departamento where nome_dept = nome_departamento and nome_cientifica = 'Engenharia';

c) select nome, salario from professor where salario < all(select salario from professor where nome_dept = 'Informatica');

d) select nome_dept, count(nome_func) as num_func, max(salario) as salario_max from professor group by nome_dept;

e) select distinct d1.edificio from departamento d1 where not exists (select 1 from departamento d2 where d2.edificio = d1.edificio and not exists (select 1 from departamento d3 where d3.edificio = d1.edificio and d3.nome_cientifica = d1.nome_cientifica));

IX

a) $(AB)^+ = ABCD$, pois todos os atributos conseguem determinar todos os outros atributos portanto é um chave candidata.

b) $(C)^+ = D$, não é chave candidata.

c) Este é um 1FN pois todos os atributos de uma relação são atômicos. Este é um 2FN pois todos os atributos não primários dependem da chave. Não está em 3FN pois existem dependências entre atributos não chave.

d) Para decompor para BCNF verificamos que $C \rightarrow D$ é funcionalmente transitiva portanto $R_1(C, D)$ e $R_2 = R - (D - C) = (A, B, C)$ e assim fica decomposto.